A Global Authentication Scheme for Mobile Ad-hoc Networks

P. Caballero-Gil[1], C. Caballero-Gil[2]

[1]Dept. Statistics, Operations Research and Computing. University of La Laguna. 38271 La Laguna. Tenerife. Spain. E-mail: pcaballe@ull.es
[2]Dept. Informatics and Systems. University of Las Palmas de Gran Canaria. 35017 Las Palmas de Gran Canaria. Gran Canaria. Spain.

Abstract. This work proposes a new global authentication system for Mobile Ad-hoc Networks. The component algorithms are designed in a self-organizing way so that most needs of this sort of networks are covered. In particular, characteristics such as adaptation to the varying topology of the network, open availability of broadcast transmissions, and strong access control have received special attention when defining the new scheme. The described protocol is based on the cryptographic paradigm of Zero-Knowledge Proofs. In this paper the design is thought for the Hamiltonian Cycle Problem, but it might be easily adapted to other NP-complete graph problems.
Keywords. Authentication, Access Control, MANETs

1 Introduction

Confidentiality, integrity and authentication are the three main security aspects that have to be taken into account when designing a secure network. Among them, authentication, which guarantees the proper identities of nodes, is the most remarkable one because the other security characteristics depend totally on the right authentication of entities.

Authentication is usually based on weak schemes of maximum-disclosure proofs with secret time-invariant passwords [14]. Their major security concern is possible eavesdropping and subsequent replay of secret passwords. Two well-known solutions to this security problem exist. The simplest of both methods uses variable passwords, whereas the strongest schemes are minimum-disclosure proofs. The protocol here proposed combines both concepts in order to define an authentication scheme specifically thought for Mobile Ad-hoc Networks.

Mobile Ad-hoc NETworks (MANETs) are autonomous networks formed by mobile nodes that are free to move at will. The set of applications for MANETs is diverse, ranging from small, static networks that are constrained by power sources, to large-scale, mobile and highly dynamic networks. Conventional wired networks mainly use a globally trusted Certificate Authority (CA) for solving the authentication problem. However, authentication in MANETs is in general

[0] Research partially supported under Project SEG2004-04352-C04-03.

much more difficult than that in wired networks due to several reasons such as limited physical protection of broadcast medium, frequent route changes caused by mobility, lack of a structured hierarchy, etc [12].

Many authentication protocols have been recently proposed for ad-hoc networks [1] [7] [11]. On the one hand, the paper [1] states the need for an authentication management architecture for ad-hoc networks. On the other hand, both papers [7] and [11] propose two different solutions. However, the first one is based on RSA signature, which conducts to the problem of public key certification, while the second solution works well just for short-lived MANETs.

In general, one of the most elementary approaches found in the bibliography uses a Trusted Third Party (TTP) to guarantee the validity of all nodes identities, so that every node who wants to join the network has to get a certificate from the TTP. A second identification paradigm that has been used in wireless ad-hoc networks is the notion of chain of trust [8]. A third typical solution is location-limited authentication, which is based on the fact that most ad-hoc networks exist in small areas and physical authentication may be carried out between nodes that are close to each other. The special nature of ad-hoc networks, where most applications are collaborative and group-based, suggests that such traditional approaches to node identification may not be always appropriate. Consequently, the design of a scheme that fulfils all the requirements for this type of networks continues being considered an open question.

This work proposes a different type of scheme based on the established cryptographic primitive of Zero-Knowledge Proofs (ZKPs), which provide an elegant solution to the problem of self-organized node authentication for MANETs. Until now very few publications have mentioned the proposal of authentication systems for ad-hoc networks using ZKPs [6] [2] [13], and none of them has dealt with the related problem of topology changes in the network. A recent ZKP-based hierarchical proposal for MANETs related with the one proposed here was described in [4], where two different security levels were defined through the use of a hard-on-average graph problem, and no topology changes were considered.

This work is organized as follows. The following section provides a complete description of the new proposal is given, including general aspects, notation and specific details about network initialization, node insertion, access control, proofs of life and node deletion.Assumptions on the scheme and security are commented in Section 3. A performance analysis is provided in Section 4. Finally, some conclusions and open questions complete the paper.

2 Proposal

The following sub-sections give, respectively, an overview of the proposal, a description of the used notation and specific details about network initialization, node insertion, access control, proofs of life and node deletion.

2.1 Overview

The proposal has been designed as an authentication scheme for group membership because when a node wants to be part of the network, it has to be previously authorized by a legitimate node. According to the authors of [10], in any group member authentication protocol it is necessary to provide robust methods to insert and to delete nodes, as well as to allow the access only for legitimate members of the group. For that reason, not only the ZKP used for access control is described later, but also the upgrade procedures associated to insertions and deletions are carefully defined. The procedure to delete nodes in this paper is based on the fact when a node is too long (according to a parameter previously agreed by the members of the network) disconnected of the network, a deletion of such a node is carried out.

The access control described below is based on the general scheme of Zero-Knowledge Proof introduced in [3], for the particular case of the Hamiltonian Cycle Problem (HCP). A hamiltonian cycle of a graph is a cycle that visits each vertex exactly once and returns to the starting vertex. Determining whether such cycles exist in a graph is the Hamiltonian Cycle Problem, which is NP-complete. Such a problem was chosen for our design mainly because the upgrade of a solution due to an insertion or a deletion of a vertex in the graph does not demand a great computational effort. Such operations will be frequent in our implementation due to the high dynamism of MANETs. Anyway, similar schemes might be described based on different NP-complete graph problems where the actualization of a solution after single changes on the graph is easy. Such is the case of Vertex Cover, Independent Set or Clique Problems, for instance.

One of the key points for the correct operation of the proposed scheme is the use of a chat application through broadcast that makes it possible for legitimate on-line nodes to send a message to all on-line users of the network. Such an application allows publishing all the information associated to the upgrade of the network. Although secrecy is not necessary for chat messages that are broadcast because they are useless for illegitimate nodes, since that information is necessary for updating authentication information, it is required that only on-line legitimate nodes of the network may execute the chat application.

The information received through the chat application during an interval of time must be stored by each on-line node in a FIFO queue. Such data stored by each on-line node allow the updating of the authentication information both for it and for all the off-line legitimate nodes whose access is authorized by that on-line node. The length of such a period is an essential parameter because it states both the maximum off-line time allowed for any legitimate node, and the frequency of broadcasts of proofs of life. Consequently, such a parameter should be previously agreed among all the legitimate nodes of the network.

A generic life-cycle of a MANET has three major phases as shown in Figure 1. Initialization is the first phase, where each initial member of the original network is securely provided, either off-line or on-line, with a secret piece of information. The knowledge of the secret network key will be used during access

control in order to prove the nodes eligibility to access protected resources or to offer service to the network.

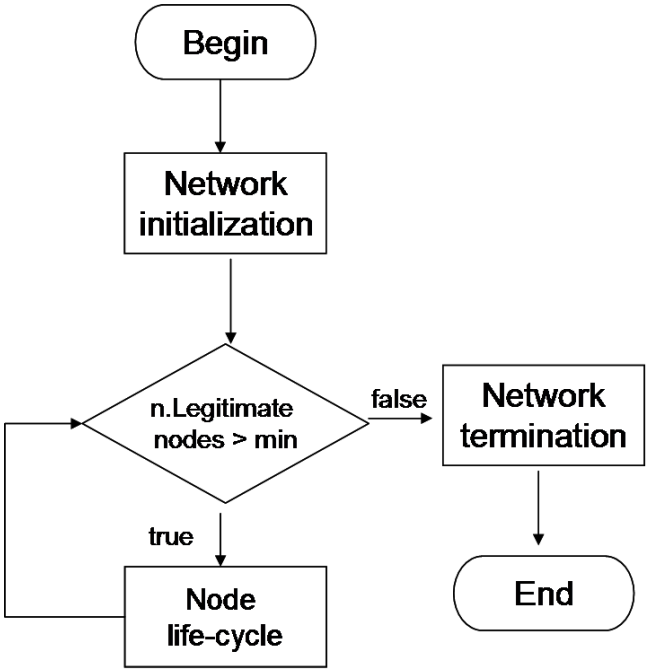

Fig. 1. Network Life-Cycle

When the initialization phase is completed, the initial legitimate nodes are ready to participate in the network, so node life-cycle starts (see Figure 2). The access control process is where a legitimate node proves its membership to an on-line node of the network. These legitimate nodes must demonstrate knowledge of the secret network key by using a challenge-response scheme.

Once the legitimate node access to the on-line state in the network is allowed, such a legitimate node gets full access to protected resources such as the chat application, and may offer services such as the insertion of new nodes. The secret network key is continuously being updated according to the development of the network, so the secret key of a legitimate node expires if this node is off-line too long. In such a case, the legitimate node would have to be re-inserted by an on-line legitimate node if it wants to enter the network again.

Since in our proposal the secrecy of the network key is provided by the difficulty of the HCP, and the number of legitimate nodes is an influential parameter in such a difficulty, as soon as the number of legitimate nodes is too small, the network termination is carried out and the life-cycle of the network ends.

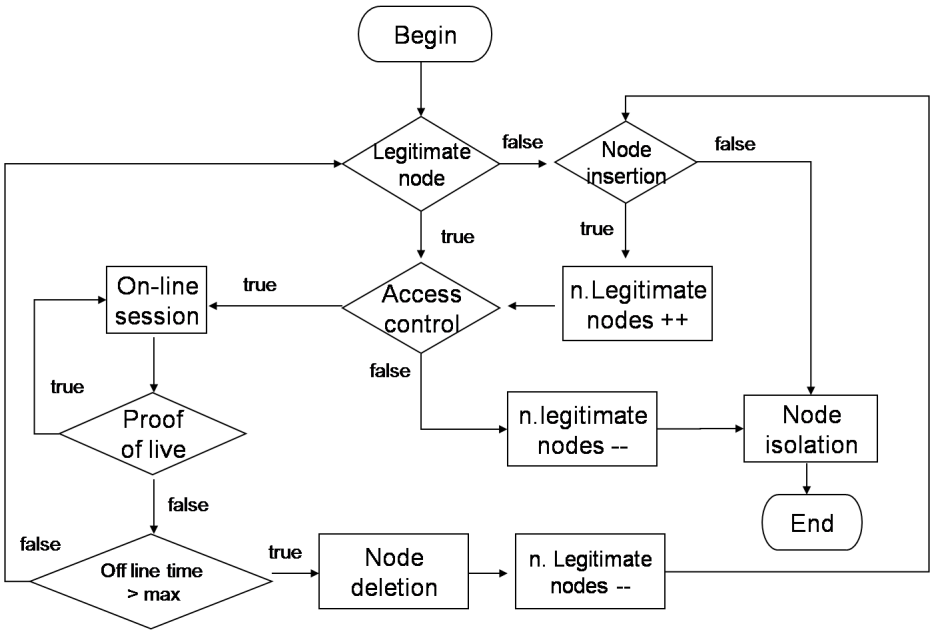

Fig. 2. Node Life-Cycle

A remarkable aspect of our proposal is that no possible adversary is able to succeed to steal any meaningful information even if it reads all the information published through the chat application, or if it eavesdrops all the information exchanged between a prover legitimate node and a verifier legitimate node during an access control.

2.2 Notation

In this section we give basic notations that are used throughout the proposal.

- $G_t = (V_t, E_t)$ denotes the undirected graph used at stage t of the network life-cycle.
- $v_i \in V_t$ represents both a vertex of the graph and a legitimate node of the network.
- $n = |V_t|$ is the order of G_t, which coincides with the number of legitimate nodes of the network.
- $N_{G_t}(v_i)$ denotes the neighbours of node v_i in the graph G_t.
- $\Pi(V_t)$ represents a random permutation over the vertex set V_t
- $\Pi(G_t)$ denotes the graph isomorphic to G_t corresponding to the permutation $\Pi(V_t)$.
- $c \in_r C$ indicates that an element c is chosen at random with uniform distribution from a set C.

- HC_t designates the hamiltonian cycle used at stage t.
- $\Pi(HC_t)$ represents the hamiltonian cycle HC_t in the graph $\Pi(G_t)$.
- $N_{HC_t}(v_i)$ denotes the neighbours of node v_i in the hamiltonian cycle HC_t.
- S and A stand for the supplicant and the authenticator, respectively, both during an insertion phase and during the execution of a ZKP-based access control.
- $S \rightleftharpoons A$ symbolizes when node S contacts A.
- $A \leftrightarrow S : information$ means that A and S agree on $information$
- $A \xrightarrow{s} S : information$ means that A sends $information$ to S through a secure channel.
- $A \xrightarrow{o} S : information$ means that A sends $information$ to S through an open channel.
- $A \xrightarrow{b} network : information$ represents when A broadcasts $information$ to all on-line legitimate nodes of the network.
- $A \xleftrightarrow{b} network : information$ represents a two-step procedure where A broadcasts $information$ to all on-line legitimate nodes of the network, and receives their answers.
- h stands for a public hash function.
- T denotes the threshold length of the off-line period for legitimate nodes.

2.3 Network Initialization

Such as it happens in most access control schemes for MANETs [5] [15], the proposed protocol requires the definition of an initialization phase where the secret information associated to the process of identification is generated and distributed within the initial network. This initialization phase consists in the definition of the graph used for the development of the protocol, jointly by all the original members of the network. Furthermore, the initialization phase implies that each legitimate member will know an initial and jointly generated solution of the HCP in such a graph.

In our proposal, as in trust graphs [9], the set of vertices of the graph corresponds exactly to the set of nodes of the real network during the whole life-cycle of the network. Consequently, the initialization process starts from a set V_0 of n vertexes corresponding to the nodes of the initial network. Furthermore, each vertex sub-index may be used as ID (IDentification) for the corresponding node. The first step of the initialization process consists of generating jointly and secretly a random permutation Π of such a set. Once this generation is completed, each legitimate node should know a hamiltonian cycle HC_0 corresponding exactly to such a permutation. Finally, the partial graph formed by the edges corresponding to such a hamiltonian cycle HC_0, is completed by adding n groups of $\frac{2m}{n}$ edges, producing the initial edge set E_0. Each one of these n groups of edges must have endvertex v_i, $i = 1, 2, ..., n$, and be randomly generated by the node v^i. Note that the cardinality $\frac{2m}{n}$ of those groups of edges must be large enough so that the cardinality of the resulting edge set $|E_0| = m$ guarantees the difficulty of the HCP in the graph G_0.

Initialization Algorithm

Input: V_0, with $|V_0| = n$

1. The n nodes of the network generate jointly, secretly and randomly the cycle $HC_0 = \Pi(V_0)$.
2. Each node $v_i \in V_0$ builds the set $N_{G_0}(i) = \{\{v_j \in_r V_0\} \cup N_{HC_0}(i)\}$ with $|N_{G_0}(i)| = \frac{2m}{n}$.
3. Each node broadcasts $v_i \xrightarrow{b} network : N_{G_0}(i)$
4. Each node merges $E_0 = \bigcup_{i=1,2,\ldots,n} \{(v_i, v_j) : v_j \in N_{G_0}(i)\}$

Output: $G_0 = (V_0, E_0)$, with $|E_0| = m$

Once the construction of the initial instance of the problem has been carried out by means of the contribution of all the nodes that are part of the network, each node will know a hamiltonian cycle in the resulting $\frac{2m}{n}$-regular graph. From then on, each time a new user S wants to become a member of the network, it has to contact a legitimate member A in order to follow the insertion procedure explained in the following section.

2.4 Node Insertion

Let us suppose that we are at stage t of the network life-cycle when a user S contacts a legitimate member A of the network to become a member of the network. Once S has convinced A to accept its entrance to the network, the first step that A should do is to assign to S the lowest vertex number v_i not assigned to any node in the vertex set V_t. Afterwards, A should broadcast such an assignment to all on-line legitimate nodes of the network in order to prevent another simultaneous insertion with the same number, and receive their answer. If A receives less than $n/2$ answers, she stops the insertion procedure because the number of nodes that are aware of the insertion is not large enough. Otherwise, A chooses the corresponding upgrade of the secret hamiltonian cycle HC_t by selecting at random two neighbour vertexes v_j and v_k in order to insert the new node v_i between them, chooses at random a set of $\frac{2m}{n} - 2$ nodes in V_t such that none of them are neighbours in HC_t, and broadcasts the set of neighbours $N_{G_{t+1}}(v_i)$ of S in the new graph G_{t+1} to all on-line legitimate nodes of the network.

Each time a node receives an updating of the graph, it should secretly update the corresponding hamiltonian cycle by using the information provided in order to identify the unique position in the cycle where the new node can be inserted according o the new edge set E_{t+1}. In this way, it will be able to easily update the secret network key by simply inserting the vertex v_i between the vertexes v_j and v_k. At the same time the authenticator node A must send the supplicant node S both the graph G_{t+1} in an open way, and the hamiltonian cycle HC_{t+1} through a secure channel.

Insertion Algorithm

Input: At stage t a supplicant node S wants to become a member of the network.

1. $S \rightleftharpoons A$ and node S convinces node A to accept its entrance to the network.
2. A assigns to S the vertex number v_i such that $i = min\{l : v_l \notin V_t\}$
3. A broadcasts $A \overset{b}{\leftrightarrow} network : v_i$
4. – If A receives less than $n/2$ answers, she stops the insertion procedure.
   – Otherwise:
   (a) A chooses at random $\{v_j \in_r V_t, v_k \in_r N_{CH_t}(v_j)\}$
   (b) A chooses at random $N_{G_{t+1}}(v_i) = \{v_j, v_k\} \cup \{w_1, w_2, ..., w_{\frac{2m}{n}-2} \in_r V_t$ such that $\forall w_{l_1}, w_{l_2} : w_{l_1} \notin N_{CH_t}(w_{l_2})\}$
   (c) A broadcasts $A \overset{b}{\rightarrow} network : N_{G_{t+1}}(v_i)$
   (d) Each on-line node computes $V_{t+1} = V_t \cup \{v_i\}$, $E_{t+1} = E_t \cup N_{G_{t+1}}(v_i)$ and $HC_{t+1} = \{HC_t \setminus (v_j, v_k)\} \cup \{(v_j, v_i) \cup (v_i, v_k)\}$
   (e) A sends openly $A \overset{o}{\rightarrow} v_i : G_{t+1}$
   (f) A sends securely $A \overset{s}{\rightarrow} v_i : HC_{t+1}$

Output: The supplicant node S is a legitimate member of the network.

2.5 Access Control

If a legitimate member of the network S that has been off-line or out-of-coverage from stage t wants to connect on-line to the network at stage r, its first step should be to contact a legitimate on-line member A. Afterwards, A should check whether the off-line period of S is not greater than T. In this case, S has to be authenticated by A through a ZKP of its knowledge of the secret solution HC_t on the graph G_t.

The aforementioned ZKP begins with the agreement between A and S on the number of iterations l of the ZKP. From there on, in each iteration, S will choose a random permutation $\Pi_j(V_t)$ on the vertex set that will be used to build a graph $\Pi(G_t)$ isomorphic to G_t. The hash value of that permutation $h(\Pi_j(V_t))$ and of the hamiltonian cycle in the graph $h(\Pi_j(HC_t))$ are then sent to A. When this information is received by A, it chooses a bit b_j at random ($b_j \in_r \{0, 1\}$). Depending on the selected value, S will provide A with the image of the hamiltonian cycle through the isomorphism, or with the specific definition of the isomorphism. In the verification phase, A will check that the received information was correctly built.

Once the authentication of supplicant S has been successfully carried out, the authenticator A gives S the necessary information to have full access to protected resources such as the chat application.

Access Control Algorithm

Input: At stage r a supplicant node S that has been off-line from stage t wants to connect on-line to the network.

– $S \rightleftharpoons A$
– S sends openly $S \overset{o}{\rightarrow} A : G_t$
– A checks whether $t \leq r - T$
  • if $t \leq r - T$ then S is not authenticated

- otherwise:
  - A and S agree $A \leftrightarrow S : l$
  - $\forall j \in \{1, 2, \ldots, l\}$
    1. S chooses $\Pi_j(V_t)$ and builds $\Pi_j(G_t)$ and $\Pi_j(V_t)$, isomorphic graph to G_t and correspondent hamiltonian cycle, respectively
    2. S sends openly $S \xrightarrow{o} A : \{h(\Pi_j(V_t)), h(\Pi_j(HC_t))\}$
    3. A chooses the challenge $b_j \in_r \{0, 1\}$
    4. A sends openly the challenge $A \xrightarrow{o} S : b_j$
       (a) If $b_j = 0$ then S sends openly $S \xrightarrow{o} A : \{\Pi_j(G_t), \Pi_j(HC_t)\}$
       (b) If $b_j = 1$ then S sends openly $S \xrightarrow{o} A : \Pi_j$
    5. A verifies
       (a) that $\Pi_j(HC_t))$ is a valid hamiltonian cycle in $\Pi_j(G_t)$, if $b_j = 0$
       (b) that the hash function h on the result of Π_j on G_t produces $h(\Pi_j(G_t))$, if $b_j = 1$
  - if $\exists j \in \{1, 2, \ldots, l\}$ such that the verification is negative, then S is isolated.
  - otherwise A sends securely $A \xrightarrow{s} S$: the necessary information to have full access to protected resources of the network.

Output: Node S is connected on-line to the network.

2.6 Proofs of Life

All on-line legitimate nodes have to confirm their presence in an active way. Such a confirmation is carried out every certain interval of time of length T so that each on-line node must broadcast a proof of life to all on-line legitimate nodes of the network.

If some insertion happens during such a period, a proof of life of every on-line legitimate node will be distributed together with the broadcast necessary for the insertion procedure. If no insertion happens during the period, the first node that has to prove its life starts a proof-of-life broadcast. During such a broadcast every node adds its own proof of life to the broadcast so that when the broadcast reaches the last node, a broadcast back starts so that when the starting node receives the proofs of life of all on-line legitimate nodes, it rebroadcasts them.

Proof-of-Life Algorithm

Input: At stage t node A is an on-line legitimate node of the network of the network.

- A initializes its $clock = 0$ just after its last proof of life
- if $clock > T$ then
  1. A broadcasts $A \xleftrightarrow{b} network : A's\ proof\ of\ life$
  2. - If A receives less than $n/2$ proofs of life as answers to her broadcast, she stops her proof of life and puts back her clock.
     - Otherwise: A broadcasts $A \xrightarrow{b} network$: $Received\ proofs\ of\ life$

Output: At stage $t + 1$ node A continues being an on-line legitimate node of the network of the network.

2.7 Node Deletion

The deletion procedure is mainly based on the confirmation of the active presence of on-line legitimate nodes through their proofs of life. Each node should update its stored graph by deleting all those nodes that have not sent any proof of life after a period T. This fact implies that each node that has not proven its life will be deleted from the network, and the corresponding vertex will be deleted from the graph and from the hamiltonian cycle.

Node deletions are explicitly communicated to all on-line legitimate nodes in the second step of broadcasts of proofs of life. In this way, any node that is off-line in that moment will be able to update its stored graph as soon as it gets access to the network.

Deletion Algorithm

Input: At stage t a node v_i is an off-line legitimate node of the network of the network.

- A initializes her $clock = 0$
- if $clock > T$ then
  1. $\forall v_i \in V_t$: A checks v_i's proof of life in A's FIFO queue
  2. A updates $V_{t+1} = V_t \setminus \{v_i \in V_t \text{ with no proof }\}$
  3. A updates $E_{t+1} = E_t \setminus \{(v_i, v_j) : v_i \in V_t \text{ with no proof, } v_j \in N_{G_t(v_i)}\} \cup \{(v_j, v_k) : v_j, v_k \in N_{HC_t(v_i)}\}$
  4. A updates $HC_{t+1} = HC_t \setminus \{(v_j, v_i), (v_i, v_k)\} \cup (v_j, v_k) : v_i \in V_t$ with no proof, $v_j, v_k \in N_{HC_t(v_i)}$
- If A was the starter of the broadcast used for the v_i's deletion, A adds this information to the second step of the proof-of-life broadcast: $A \xrightarrow{b} network : v_i$ is deleted.

Output: At stage $t + 1$ the node v_i has been deleted both from the network and from the graph.

This way to proceed guarantees a limited growth of the graph that is used in authentication, and at the same time, allows that always legitimate nodes of the network correspond exactly to vertexes in that graph. Apart from this, it is remarkable the fact that thanks to this procedure the recovery of legitimate members of the network that have been disconnected momentarily due to a shortcut of the network is possible, if such a shortcut does not last too much (i.e. if it is lesser than T).

3 Assumptions and Security Analysis

This proposal assumes initially the ideal environment where all legitimate nodes are honest and where no adversary may compromise a legitimate node of the network in order to read its secret stored information. Such assumptions are well suited as a basic model in order to decide under which circumstances the designed authentication scheme is applicable to MANETs. For instance, a possible adaptation of the proposal in order to avoid those hypothesis could be the

consideration of a threshold scheme for every step of the scheme, so that every proof of life, insertion, access control or deletion should be done by a group of on-line nodes each time. In this way, a dishonest node would not affect the correct operation of the network.

It is also clear that the proposal inherits inherent problems of the distributed trust model such as the important necessity that legitimate nodes cooperate. Consequently, it is advisable that some scheme to stimulate node cooperation is used in conjunction with the proposal.

Finally, another requirement of the scheme is the necessary establishment of a secure channel for the insertion procedure. However, that aspect may be easily fulfilled thanks to the fact that most wireless devices communicate with each other via Bluetooth wireless technology.

With respect to possible attacks, due to the lack of a centralized structure, it is natural that possible DOS (Denial Of Service) attacks have as their main objective the chat application. In order to protect the scheme against this threat it must be assured that chat messages, although are publicly readable, may be only sent by legitimate on-line members of the network. Another important aspect related to the use of the chat application is the necessary synchronization of the on-line nodes, so a common network clock is necessary. this requirement has been implemented during simulations through the chat application.

MANETs are in general vulnerable to different threats such as identity theft (spoofing) and the man-in-the-middle attack. Such attacks are difficult to prevent in environments where membership and network structure are dynamic and the presence of central directories cannot be assumed. However, our proposal is resistant to spoofing attacks because access control is proved through a ZKP that makes useless the reading of any information published through the chat application or sent openly during an access control. On the other hand, the goal of the man-in-the-middle attack is either to change a sent message or to gain some useful information by one of the intermediate nodes. Again the use of ZKPs in our protocol implies that reading any transferred information does not reveal any useful information about the secret, so changing the message is not possible since only legitimate nodes whose access has been allowed can use the chat application.

Another active attack that might be especially dangerous in MANETs is the so-called Sybil attack. It happens when a node tries to get and use multiple identities. The most extreme case of this type of attacks is the establishment of a false centralized authority who states the identities of legitimate members. However, this specific attack is not possible against our scheme due to its distributed nature. In our scheme, the responsibility of controlling general Sybil attacks will be shared among all the on-line nodes. If an authenticator node detects that a supplicant node is trying to get access to the network by using an ID that is yet being used on-line, such access control must be denied and the corresponding node must be isolated. The same happens when any on-line node detects that an authenticator node is trying to insert a new node to the network with a new ID, and such a node has yet assigned a vertex ID. Again, such insertion must

be denied and the corresponding supplicant node must be isolated. Anyway, if a Sybil attacker enters the network, any of its neighbours will detect it as soon as it sends proofs of life for different vertexes ID.

4 Performance Analysis

We now analyze the efficiency of the proposal both from the energy consumption and from computational complexity points of view. We consider the energy consumption which is the result of transmissions of data and processor activities due to authentication tasks. In the proposal there are two phases when computational overhead is more significant: the ZKP-based access control and the periodic checking of stored FIFO queue. A reduction on the number of rounds of ZKP has a direct effect on the total exchanged messages size in insertions, but a trade-off should be maintained between protocols robustness and performance. Indeed, regarding total data transmission over wireless links, the ZKPs take less than 10% in a usual situation.

The periodic proofs of life accounts for approximately 90% of the total exchanged message size in many cases. However, we have found that these compulsory proofs of life imply an incentive technique for stimulating cooperation in authentication tasks. This is due to the fact that nodes that are broadcasters of deletions or authenticators in insertions or access controls are exempted from their obligation to broadcast their proofs of life.

In order to reduce data communication cost of the protocol, an increase on the threshold period T might be an option, but again an acceptable balance should be kept. According to our experiments, T should depend directly on the number of legitimate and/or on-line nodes in order to prevent a possible bandwidth overhead of large networks.

For the performance analysis of the proposal we used the Network Simulator NS-2 with DSR routing protocol. We created several Tcl based NS-2 scripts in order to produce various output trace files that have been used both to do data processing and to visualize the simulation. Within our simulation we used the visualization tool of Network Animator NAM and the NS-2 trace files analyzer of Tracegraph. For the simulation of mobility we used the setdest program in order to generate movement pattern files using the random waypoint algorithm.

An example of simulation is shown graphically in Figure 3. Basically it consists of generating a scenario file that describes the movement pattern of the nodes and a communication file that describes the traffic in the network. These files are used to produce trace files that are analyzed to measure various parameters. An excerpt of the trace files corresponding to the same example is shown in Table 1.

The trace files are used to visualize the simulation using NAM, while the measurement values are used as data for plots with Tracegraph. The final graph and hamiltonian cycle associated to the example network is shown in Figure 4 where green is used to indicate the hamiltonian cycle, blue is used for the

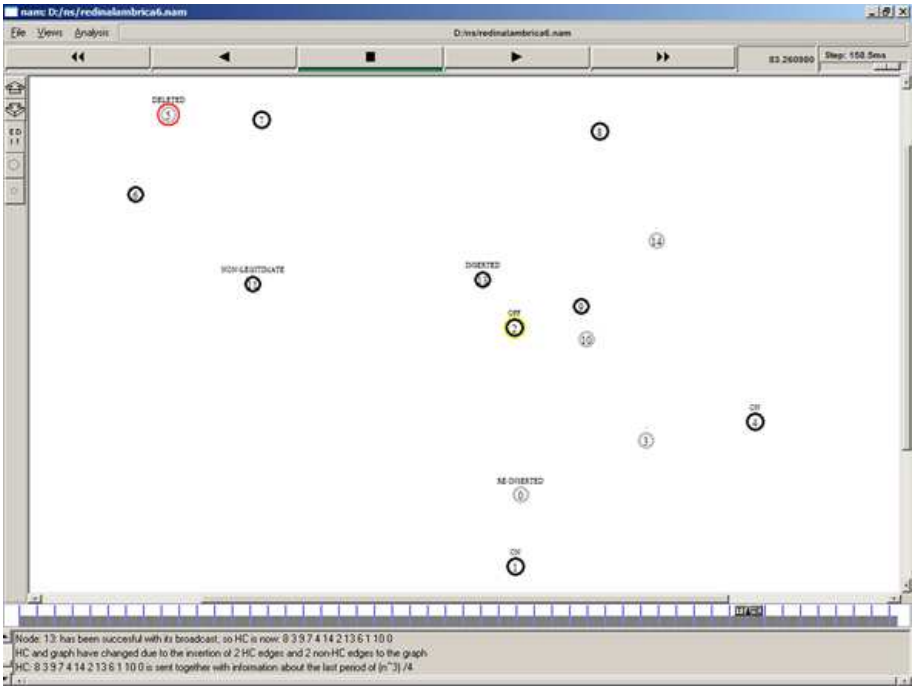

Fig. 3. Example of Network Simulation with NS-2

inserted nodes and red is used for the edges deleted from the hamiltonian cycle when inserting new nodes.

We conducted many different simulations in order to see the effects of different metrics by varying network density and topology. In particular, we varied the number of nodes from 15 to 100, the area from 400x400 to 800x800 m^2, and the period of simulation from 60 to 200 seconds. We also changed the probabilities of insertions and deletions in each second from 5% to 25%, in order to modify the mobility rate and antenna range of nodes from 2 to 15 m/s and 100 to 250 meters respectively. This range also defines different frequencies of accesses to the network.

The first conclusions that we have obtained from the simulations are:

- The protocol scales perfectly to any sort of networks with different levels of topology changes.
- Node density is a key factor for the mean time of insertions, but such a factor is not as big as it might be previously assumed since nodes do not forward two packets of data corresponding to the same proof of life coming from two different nodes.

Time	Event	H.C.
0.1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 are legitimate	8,3,9,7,4,2,6,5,1,10,0
1.29	Insertion of Node 14 is broadcast by Node 4	8,3,9,7,4,14,2,6,5,1,10,0
1.30	Nodes 3, 1, 0 do not answer to the proof of life	
3.29	Node 0 reaches 8 and starts a ZKP for re-insertion	
8.69	Node 3 reaches 4 and starts a ZKP for re-insertion	
9.40	Node 1 reaches 10 and starts a ZKP for re-insertion	
11.65	Node 1 turns off	
13.97	Proof of life started by Node 3	
14.27	Nodes 1, 2 do not answer to the proof of life	
14.82	Node 2 reaches 14 and starts a ZKP for re-insertion	
17.27	Proof of life started by Node 2	
17.57	Nodes 1, 5 do not answer to the proof of life	
21.71	Node 5 turns off	
31.40	Node 1 turns on and Node 2 is chosen for the ZKP	
31.46	Node 4 turns off	
32.51	Proof of life started by Node 1	
32.78	Nodes 4, 5, 6 do not answer to the proof of life	
34.29	Node 6 reaches 2 and starts a ZKP for re-insertion	
38.51	Proof of life started by Node 6	
38.79	Nodes 4, 5 do not answer to the proof of life	
41.46	Node 1 turns off	
53.25	Node 1 turns on and Node 0 is chosen for the ZKP	
59.61	Proof of life started by Node 6	
59.99	Nodes 4, 5 do not answer to the proof of life	
64.26	Node 5 is deleted	8,3,9,7,4,14,2,6,1,10,0
64.71	Node 2 turns off	
72.58	Node 4 turns on and Node 0 is chosen for the ZKP	
75.41	Insertion of Node 13 is broadcast by Node 14	8,3,9,7,4,14,2,13,6,1,10,0
75.43	Node 2 does not answer to the proof of life	

Table 1. Example of Trace

- A right choice of parameter T should be done according to number of nodes, bandwidth of wireless connections and computation and storing capacities of nodes.
- A positive aspect of the proposal is that the requirements in the devices' hardware are very low.

5 Conclusions and Open Questions

This work describes a new authentication scheme that has been specially designed for MANETs. Such a protocol supports knowledge-based member authentication in server-less environments. The overall goal of this proposal has been to design a strong authentication scheme that is able to react and adapt to network topology changes without the necessity of any centralized authority. Its

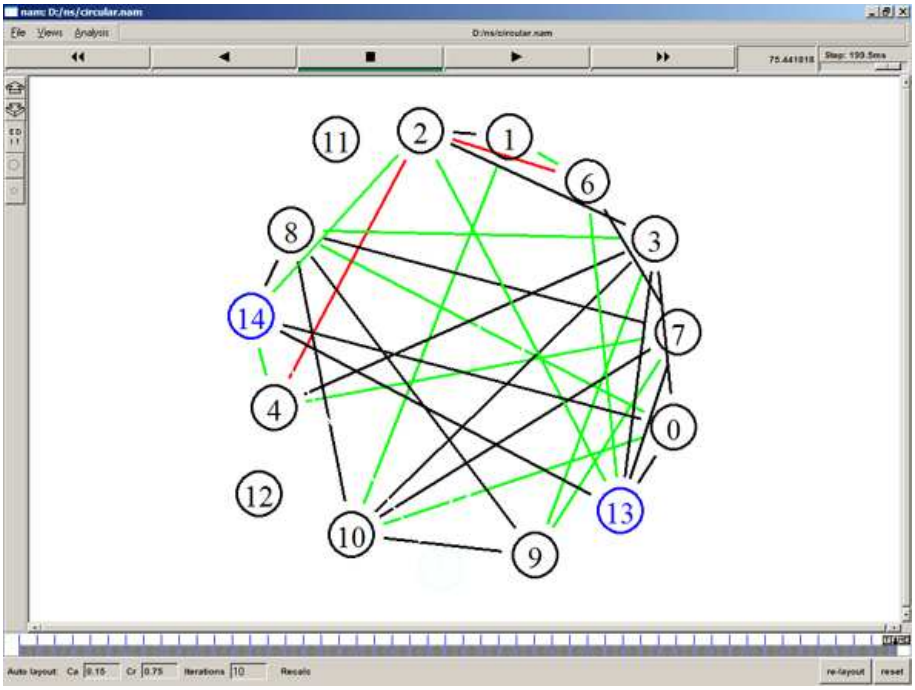

Fig. 4. Example of Final Associated Graph and Hamiltonian Cycle

core technique consists of a Zero-Knowledge Proof, in order to avoid the transference of any relevant information. Furthermore, the proposal is balanced since the procedures that the legitimate members of the network have to carry out when the network is upgraded (insertion or deletion of nodes) imply identical work for every legitimate member of the network.

The development of an initial simulation of the proposal through the NS-2 network simulator has been carried out. The definitive simulation results will be included in a forthcoming version of this work. Also, the study of different applications, practical limitations and possible extensions of the proposed scheme may be considered open problems.

References

1. Aboudagga, N., Tamer, M., Eltoweissy, M., DaSilva, L. and Quisquater,J.J.: Authentication protocols for ad hoc networks: Taxonomy and research issues, Proceedings of the 1st ACM international workshop on Quality of service and security in wireless and mobile networks, October (2005)
2. Asaeda, H., Rahman, M., Manshaei, H., and Fukuzawa, Y.: Implementation of Group Member Authentication Protocol in Mobile Ad-hoc Networks, Proceedings

of IEEE Wireless Communications and Networking Conference WCNC, Las Vegas, USA April (2006)
3. Caballero-Gil, P., Hernández-Goya, C.: Strong solutions to the identification problem. Proceedings of COCOON. Lecture Notes in Computer Science Vol. 2108, Springer-Verlag (2001) 257-261
4. Caballero-Gil, P., Hernández-Goya, C.: Zero-Knowledge Hierarchical Authentication in MANETs. IEICE Transactions on Information and Systems. Letter. E-89-D (2006) 1288-1289
5. Capkun, S., Buttyan, L., Hubaux, J.P.: Self-organized public-key management for mobile ad-hoc networks. IEEE Transactions on Mobile Computing (2003)
6. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. Proceedings of Crypto '86, Lecture Notes in Computer Science Vol. 263. Springer-Verlag (1986) 171-185
7. Hahm, S., Jung, Y., Yi, S., Song, Y., Chong, I. and Lim, K.: A Self-Organized Architecture in Mobile Ad-Hoc Networks. Proceedings of ICOIN, Lecture Notes in Computer Science Vol. 3291 Springer-Verlag (2005) 689-696
8. Hubaux, J.P., Buttyán, L., Capkun, S.: The quest for security in mobile ad hoc networks. Proceedings of MobiHoc. (2001) 146-155
9. Jiang, T. and Baras J.S.: Graph Algebraic Interpretation of Trust Establishment in Autonomic Networks. Submitted to Wiley Journal of Networks, May 2005, under review
10. Maki, S., Aura, T., Hietalathi, M.: Robust membership management for ad-hoc groups. Proceedings of 5th Nordic Workshop on Secure IT Systems NORDSEC (2000)
11. Saxena, N., Tsudik, G., Yi, J.H.: Efficient node admission for short-lived mobile ad hoc networks. IEEE International Conference on Network Protocols ICNP, November (2005) 269-278
12. Weimerskirch, A.: Authentication in Ad-hoc and Sensor Networks. Ph.D. Thesis Ruhr-University Bochum. Germany, July (2004)
13. Wierzbicki, A. , Zwierko, A. and Kotulski, Z.: A New Authentication Protocol for Revocable Anonymity in Ad-Hoc Networks. Proceedings of the IASTED Communication, Network, and Information Security CNIS, Phoenix, AZ, USA (2005)
14. Wu, H-C., Hwang, M-S. and Liu, C-H.: A Secure Strong-Password Authentication Protocol. Fundamenta Informaticae 68 (2005) 399-406
15. Zhou, L., Haas, Z.: Securing ad hoc networks. IEEE Networks 13 (1999) 24-30